UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/501,845	08/26/2005	Martin Vorbach	2885/86	9148

26646 7590 12/29/2011
KENYON & KENYON LLP
ONE BROADWAY
NEW YORK, NY 10004

EXAMINER	
VICARY, KEITH E	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
12/29/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/501,845	**Applicant(s)** VORBACH ET AL.	
	Examiner KEITH VICARY	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _08 September 2011_.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) _7-13,15-17 and 19-33_ is/are pending in the application.

5a) Of the above claim(s) _23 and 28-31_ is/are withdrawn from consideration.

6)☒ Claim(s) _25-27,32 and 33_ is/are allowed.

7)☒ Claim(s) _7-13,15-17,19-22 and 24_ is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

12)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _See Continuation Sheet_.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :9/8/2011, 11/3/2011 (first), 11/3/2011 (second), 11/28/2011.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 9/8/2011 has been entered.

2. Claims 7-13, 15-17, and 19-33 are pending in this office action. Claims 7-8, 10-12, 15, 17, 19-20, and 23-24 are newly amended and claims 25-33 are newly added by amendment filed 9/8/2011. Claims 23 and 28-31 are withdrawn from consideration as being directed to a non-elected invention. Consequently, claims 7-13, 15-17, 19-22, 24-27, and 32-33 are presented for examination.

Election/Restrictions

3. Newly submitted claims 23 and 28-31 are directed to an invention that is independent or distinct from the invention originally claimed for the following reasons:

With regard to claims 23 and 28, claim 23 recites of "modifying, by a processor executing a software compiler, a loop of a program, the program including a sequence of compilable high-level language instructions, to generate a modified version of the program, wherein the modifying of the loop includes inserting into the loop a maximum

value condition that is based on a maximum allowed execution runtime of a configuration of at least a subset of the data processing cells; and outputting the modified version of the program." The concept of a software compiler generating a modified version of a program by modifying a loop in an original program is independent or distinct from the concept of "[a] processor, comprising: a reconfigurable field of data processing cells; and a register, wherein the register has a data stream memory designed to store at least one of a data stream and parts of the data stream" as originally claimed, and of "for each configuration, monitoring the respective maximum allowed execution runtime in order to interrupt processing with the configuration if the respective maximum allowed execution runtime is exceeded" as more recently claimed in conjunction with the aforementioned processor of the original set of claims. Claim 28 is dependent on claim 23 and is likewise independent or distinct from the invention originally claimed for the same reasons.

With regard to claims 29-31, claim 29 recites of "a method of data processing using a processor comprising a reconfigurable field of data processing cells, the method comprising: while the field is configured with a first configuration, triggering, with the first configuration, extraction of an entry from a stack and transmission of the entry as a call for reconfiguration." The concept of "while the field is configured with a first configuration, triggering, with the first configuration, extraction of an entry from a stack and transmission of the entry as a call for reconfiguration" is independent or distinct from the concept of "[a] processor, comprising: a reconfigurable field of data processing cells; and a register, wherein the register has a data stream memory designed to store

at least one of a data stream and parts of the data stream" as originally claimed, and of "for each configuration, monitoring the respective maximum allowed execution runtime in order to interrupt processing with the configuration if the respective maximum allowed execution runtime is exceeded" as more recently claimed in conjunction with the aforementioned processor of the original set of claims. Claims 30-31 are dependent on claim 29 and are likewise independent or distinct from the invention originally claimed for the same reasons.

Since applicant has received an action on the merits for the originally presented invention, this invention has been constructively elected by original presentation for prosecution on the merits. Accordingly, claims 23 and 28-31 are withdrawn from consideration as being directed to a non-elected invention. See 37 CFR 1.142(b) and MPEP § 821.03.

Claim Objections

4. Claims 7-13, 15-17, 19-22, and 24 are objected to because of the following informalities. Appropriate correction is required.

5. In claim 7, line 23, "the configuration" should be "a configuration" for antecedent basis purposes.

6. In claim 24, line 24, "the bus structure" should be "a bus structure".

   a. Claims 8-13, 15-17, and 19-22 are objected to for failing to alleviate the objection of claim 24 above.

Claim Rejections - 35 USC § 112

7. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

8. Claim 7 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

9. Claim 7 recites the limitation "responsive to determining in the monitoring step that the respective maximum allowed execution runtime is exceeded, interrupting execution" in lines 27-29. However, claim 7 also discloses of "configuring at least a subset of the data processing cells for execution by the at least the subset of the data processing cells of an entire loop of a program" in lines 18-20. The original disclosure does not appear to support the embodiment wherein both "configuring at least a subset of the data processing cells for execution by the at least the subset of the data processing cells of an entire loop of a program" and "responsive to determining in the monitoring step that the respective maximum allowed execution runtime is exceeded, interrupting execution" occurs. For example, page 16, lines 12-14, appears to indicate that "responsive to determining in the monitoring step that the respective maximum allowed execution runtime is exceeded, interrupting execution" (e.g. executing loop control via reconfiguration) occurs only in the embodiment wherein the loop body is

translated into a configuration, and not the embodiment wherein both the loop body and the loop control (in other words, the entire loop) are translated into a configuration (as, in this latter embodiment, the "for loop" appears to control when the configuration is finished executing, and while this "for loop" entails the use of the maximum running time, it does not entail monitoring that the respective maximum allowed execution runtime is exceeded; see page 17, lines 15-19).

10. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

11. Claim 7 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

12. Claim 7 recites the limitation "an entire loop of a program" in line 30. It is indefinite as to whether an "entire" loop of a program means all instructions of a loop control, all instructions of a loop body, or all instructions in both a loop control and a loop body.

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

14. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Smith et al. (Smith) (US 6658564 B1) in view of Li et al. (Li) (Hardware-Software Co-Design of Embedded Reconfigurable Architectures) in view of Borkenhagen et al. (Borkenhagen) (US 6076157).

15. Consider claim 7, Smith discloses a method of data processing using a processor comprising a reconfigurable field of data processing cells (col. 3, lines 2-3, array of programmable logic regions), the method comprising: configuring at least a subset of the data processing cells for execution by the at least the subset of the data processing cells of a portion of a program (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions), the program including a sequence of compilable high-level language instructions (col. 10, lines 49-50, high-level design specification or algorithm); during the execution of the portion of the program using the at least the subset of the data processing cells according to the configuration established by the configuring step (col. 10, lines 50-51, executing on a reconfigurable hardware architecture), and reconfiguring one or more of the at least the subset of the data processing cells (col. 6, lines 53-56, the virtual logic manager must therefore manage the run-time swapping of functions to be implemented in programmable logic).

However, Smith does not disclose that the configuration is of an entire loop. Smith also does not disclose of monitoring a respective maximum allowed execution

runtime of the configuration, and responsive to determining in the monitoring step that the respective maximum allowed execution runtime is exceeded, interrupting execution of the loop prior to its completion, wherein the interrupting includes reconfiguring one or more of the at least the subset of the data processing cells.

On the other hand, Li discloses that a configuration is of an entire loop (section 1, fifth paragraph, lines 4-9, each loop has a software version and one or more hardware versions that represent different delay and area tradeoffs. The partitioning algorithm selects which loops to implement in the FPGA, and which hardware version of each loop should be used to achieve the highest application-level performance).

LI's teaching improves system performance (Li, section 3.1, third paragraph, lines 1-5, our studies show that loops represent a significant portion of application execution time, and yet are usually compact enough to implement in modest hardware resources. Therefore, the compiler focuses on finding the most profitable loops to extract out as hardware kernels).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Li with the invention of Smith in order to improve system performance.

However, Smith and Li do not disclose of monitoring a respective maximum allowed execution runtime of the configuration, and responsive to determining in the monitoring step that the respective maximum allowed execution runtime is exceeded, interrupting execution of the loop prior to its completion, wherein the interrupting

includes reconfiguring one or more of the at least the subset of the data processing cells.

On the other hand, Borkenhagen discloses of monitoring a respective maximum allowed execution runtime of a thread, and responsive to determining in the monitoring step that the respective maximum allowed execution runtime is exceeded, interrupting execution of the thread prior to its completion (col. 15, lines 1-7, thread switch time-out value is decremented, and when it is determined to equal zero, a thread switch is forced).

Borkenhagen's teaching of interrupting after the maximum allowed execution runtime prevents processor hangs (Borkenhagen, col. 5, lines 35-37).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Borkenhagen with the invention of Smith and Li in order to prevent processor hangs in a flexible manner while enabling forward progress. Note that Borkenhagen's teaching of monitoring a respective maximum allowed execution runtime of a thread, and responsive to determining in the monitoring step that the respective maximum allowed execution runtime is exceeded, interrupting execution of the thread prior to its completion, when applied to the combination of Smith (wherein configurations can be used instead of threads and configurations are switched by reconfiguring one or more of the at least the subset of the data processing cells) and Li (wherein configurations implement entire loops), results in the overall claimed limitation of monitoring a respective maximum allowed execution runtime of the configuration, and responsive to determining in the monitoring

step that the respective maximum allowed execution runtime is exceeded, interrupting execution of the loop prior to its completion, wherein the interrupting includes reconfiguring one or more of the at least the subset of the data processing cells.

16. Claims 10-11, 15-17, 19-22, and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith et al. (Smith) (US 6658564 B1) in view of Dockser (US 5860119) in view of Borkenhagen et al. (Borkenhagen) (US 6076157) in view of Bondalapati et al. (Bondalapati) (Reconfigurable Meshes: Theory and Practice).

17. Consider claim 24, Smith discloses a method of data processing using a processor comprising a reconfigurable field of data processing cells (col. 3, lines 2-3, array of programmable logic regions); the method comprising: providing a program corresponding to a sequence of compilable high-level language instructions (col. 10, lines 49-50, high-level design specification or algorithm); providing a program corresponding to a sequence of compilable high-level language instructions (col. 10, lines 49-50, high-level design specification or algorithm); determining, for the reconfigurable field of data processing cells (col. 8, lines 52-53, programmable logic resources), a set of configurations of the reconfigurable field of data processing cells, with respect to at least one of a function and an interconnection of the reconfigurable field of data processing cells, with execution of which configurations the program is run (col. 11, lines 60-63, compiling hardware functions into configuration patterns using a hardware description language compiler), wherein, for each of at least one of the configurations, a plurality of instructions of the program are executable via a single

instance of the respective configuration (col. 13, lines 33-34, a single block of configuration data that makes up a given function); a software compiler (col. 11, lines 62-63, hardware description language compiler 84); executing the configurations (col. 10, lines 50-51, executing on a reconfigurable hardware architecture); and during the executing: configuring functions of at least a subset of the data processing cells (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions); configuring an interconnection of at least a subset of the data processing cells (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions; the routing structure is the interconnection); storing, in a data stream memory, at least one of the data stream and parts of the data stream (col. 4, lines 22-33, disclose of the random-access memory devices, it is inherent that they may be written to, the data with which it is written constitutes all or part of a data stream), wherein the data stream memory stores at least one data vector (it is inherent that a data stream memory holds vectors of bits, such as each addressable line).

However, Smith does not explicitly disclose that the data stream memory is a register which is operated as a FIFO memory. Smith also does not disclose determining, by a software compiler and for each of the configurations, a respective

maximum allowed execution runtime prior to lapse of which the respective configuration is uninterruptible, and for each configuration, monitoring the respective maximum allowed execution runtime in order to interrupt processing with the configuration if the respective maximum allowed execution runtime is exceeded. Smith also does not disclose of dynamically allocating resources of the bus structure.

On the other hand, Dockser does disclose a register which is operated as a FIFO memory to process a data stream (e.g. col. 12, line 16-17, FIFO registers).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that one of many motivations of having a register act as a memory would be to allow quick access to data, and a FIFO buffer preserves order of incoming data (Dockser, col. 1, lines 63-65). In addition, Dockser makes a FIFO system both simple and inexpensive to implement (Dockser, col. 4, lines 6-40), despite decreases in management overhead.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Dockser with the invention of Smith in order to allow quicker access to the data stream while preserving order of incoming data in a simple and inexpensive manner.

However, neither Smith nor Dockser disclose determining, by a software compiler and for each of the configurations, a respective maximum allowed execution runtime prior to lapse of which the respective configuration is uninterruptible, and for each configuration, monitoring the respective maximum allowed execution runtime in order to interrupt processing with the configuration if the respective maximum allowed

execution runtime is exceeded. Smith and Dockser also do not disclose of dynamically allocating resources of the bus structure.

On the other hand, Borkenhagen does disclose of determining, by software (col. 15, lines 20-22, the thread switch time-out register can be written by the processor itself with software code) and for each thread (col. 14, lines 65-67, each thread need not have the same thread switch time-out value), a respective maximum allowed execution runtime (col. 15, line 1, thread switch time-out value) prior to lapse of which a respective thread is uninterruptible (col. 14, line 45-48 discloses of the use of the time-out register so that an external interrupt is serviced within a limited period of time; thus the interrupt is not serviced until the time-out register indicates a time-out and not before), and for each thread, monitoring the respective maximum allowed execution runtime in order to interrupt the thread if the respective maximum allowed execution runtime is exceeded (col. 15, lines 1-7, thread switch time-out value is decremented, and when it is determined to equal zero, a thread switch is forced).

Borkenhagen's teaching of interrupting after the maximum allowed execution runtime prevents processor hangs (Borkenhagen, col. 5, lines 35-37). It would also be readily recognized that the use of different thread switch time-out values for different threads increases flexibility, and interrupting only after the time-out enables forward progress.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Borkenhagen with the invention of Smith and Dockser in order to prevent processor hangs in a flexible manner while

enabling forward progress. Note that the overall combination results in Borkenhagen's specific thread switching policy being applied to Smith's configurations (and Smith's generic configuration switching policy) to result in an overall specific configuration switching policy, and Borkenhagen's teaching of using software to determine a thread-switch time-out value being applied to Smith's invention which uses a software compiler to result in the determination by a software compiler of a respective maximum allowed execution runtime.

However, Smith, Dockser, and Borkenhagen do not disclose of dynamically allocating resources of the bus structure.

On the other hand, Bondalapati discloses of dynamically allocating resources of the bus structure. (e.g. page 2, FPGA devices; page 9, section 4, lines 1-3, the reconfigurable mesh has nearly constant diameter and a dynamically reconfigurable bus system. It is very attractive in terms of implementation because of the two dimensional topology, low-pin requirement and highly regular structure).

Bondalapati's teaching is very attractive in terms of implementation because of the two dimensional topology, low pin requirement, and highly regular structure (Bondalapati, page 9, section 4, lines 1-3, the reconfigurable mesh has nearly constant diameter and a dynamically reconfigurable bus system. It is very attractive in terms of implementation because of the two dimensional topology, low-pin requirement and highly regular structure).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Bondalapati with the invention of Smith,

Dockser, and Borkenhagen in order to be very attractive in terms of implementation because of the two dimensional topology, low pin requirement, and highly regular structure.

18. Consider claim 10, Smith discloses that the register is a RAM PAE (col. 4, lines 22-33, disclose of the random-access memory devices).

19. Consider claim 11, the combination of Smith, Dockser, and Borkenhagen discloses using the register to provide read and write access (Dockser, col. 4, lines 32-35, receive mode and transmit mode, and col. 5, lines 56-65, read and write pointers) when a virtual FIFO dividing line is implemented (Dockser, col. 3, lines 10-30, lines 54-56; the last word flag and end-of-packet detection means correlate to the said virtual FIFO dividing line), wherein the program includes a multitask application, and a register is used for execution of at least one of two different tasks of the multitask application (Smith and Borkenhagen discloses of multitasking as cited in the independent claim; alternatively, multiple tasks within a thread).

20. Consider claim 15, Borkenhagen discloses a watchdog is used to recognize an exceedance of each respective maximum allowed execution runtime (col. 15, lines 1-3, thread switch time-out values from the thread switch time-out register forces a thread switch).

21. Consider claim 16, Borkenhagen discloses that any one of the configurations that exceeds its respective maximum allowed execution runtime is treated as illegal (col. 15, lines 1-3, thread switch time-out values from the thread switch time-out register forces a thread switch; thus it is illegal for the first thread to continue executing).

22. Consider claim 17, Borkenhagen discloses that any one of the configurations that exceeds its respective maximum allowed execution runtime is treated as illegal (col. 15, lines 1-3, thread switch time-out values from the thread switch time-out register forces a thread switch; thus it is illegal for the first thread to continue executing).

23. Consider claim 19, Smith and Borkenhagen discloses an operating system performs a predefined step in response to an exceedance by a configuration of the configuration's maximum allowed execution (Borkenhagen, col. 15, lines 1-19, thread switch time-out values forces a thread switch; col. 17, lines 41-42, operating system; Smith discloses in col. 8, lines 26-27, 52-53, discloses of operating systems allocating programmable logic resources to functions).

24. Consider claim 20, Smith discloses at least one of the configurations calls another of the configurations as a sub-routine (col. 12, lines 1-5 for example, a main function calls a dynamically-linked function).

25. Consider claim 21, Borkenhagen discloses the watchdog signal initiates a system trap (col. 15, lines 1-3, thread switch time-out values from the thread switch time-out register forces a thread switch; in other words, the thread is interrupted by the thread switching system).

26. Consider claim 22, Smith and Borkenhagen discloses, in response to the system trap, an operating system performs steps defined for a response to an invalid instruction (Borkenhagen, col. 15, lines 1-19, thread switch time-out values forces a thread switch; col. 17, lines 41-42, operating system; Smith discloses in col. 8, lines 26-27, 52-53, discloses of operating systems allocating programmable logic resources to functions).

27. Claims 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith, Dockser, Borkenhagen, and Bondalapati as applied to claim 24 above, and further in view of Panwar et al. (Panwar) (US 5941977).

28. Consider claim 8, Smith, Dockser, Borkenhagen, and Bondalapati do not disclose at least one: I) of a register allocation device to allocate the register, and ii) a register releasing device to release the register.

On the other hand, Panwar does disclose at least one: I) of a register allocation device to allocate the register, and ii) a register releasing device to release the register (col. 7, lines 31-39, register window allocation and col. 7, lines 54-64, register management).

Panwar's teaching enables processes to access registers independent of other processes executing within the processor (Panwar, col. 7, lines 35-38).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Panwar with the invention of Smith, Dockser, Borkenhagen, and Bondalapati in order to access registers independent of other processes executing within the processor.

29. Consider claim 9, Panwar discloses that the register allocation device is preserved over multiple reconfigurations of the reconfigurable field of data processing cells (col. 2, lines 25-42, col. 6, lines 32-36, col. 7, lines 31-39 and 54-64; the multithreading aspect in which each thread has its corresponding registers conserved correlates to the different reconfigurations as per Smith's correlation between configurations and threads as explained above).

30. Claims 12 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith, Dockser, Borkenhagen, and Bondalapati as applied to claim 24 above, and further in view of Davis et al. (Davis) (US PAT 4041462).

31. Consider claim 12, Smith, Dockser, Borkenhagen, and Bondalapati do not explicitly disclose at least one memory unit configured for use as a stack and being configured to indicate at least one of a stack underflow state and a stack overflow state.

On the other hand, Davis does disclose at least one memory unit configured for use as a stack and being configured to indicate at least one of a stack underflow state

and a stack overflow state (col. 14, lines 1-4, limit checking facilities which test for overflow and underflow, and lines 21-32, PSW)

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that stacks in general are an easily implemented method of dynamic allocation of storage space for data, and a simple efficient mechanism for enqueuing data and/or parameters.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the stacks of Davis with the invention of Smith, Dockser, Borkenhagen, and Bondalapati in order to easily implement a method of dynamic allocation of storage space for data, and efficiently enqueue data and/or parameters.

32. Consider claim 13, the claim is rejected for the same reasons as claim 12 above. In addition, Davis discloses the at least one of the underflow state and overflow state is of an operating system unit (col. 14, lines 1-4 and lines 21-32; also, note the PSW is typically accessed by the operating system).

Allowable Subject Matter

33. Claims 25-27 and 32-33 are allowed.

34. The following is a statement of reasons for the indication of allowable subject matter.

With regard to claim 25, the prior art of record, alone or in combination, does not disclose “responsive to determining, in the monitoring step, that the maximum allowed execution runtime of the first configuration is exceeded, removing, by the processor, the first configuration and the one or more other configurations" in the context of and in combination with the remaining limitations of the claim.

With regard to claim 32, the prior art of record, alone or in combination, does not disclose “responsive to determining in the monitoring step that the maximum allowed execution runtime is exceeded, removing the first configuration and configuring the field with a second configuration prior to readout of all of the data elements of the data vector, such that a second subset of the data elements of the data vector remains unread in the memory arrangement; and subsequent to the removing of the first configuration and the configuring of the field with the second configuration, sequentially reading, by the field using the second configuration, one or more data elements of the second subset of the data elements” in the context of and in combination with the remaining limitations of the claim.

Response to Arguments

35. Applicant on page 11 first appropriately characterizes the Borkenhagen reference as referring to a maximum runtime for a thread, and not for a configuration. Applicant then argues across pages 11-12 that Smith's teaching of a configuration and of a thread would not suggest applying Borkenhagen's teaching of a maximum runtime to a configuration; this argument is addressed below.

36. Applicant argues on page 12 that, thus, there is no one-to-one correspondence of a thread to a configuration, and the reference to a forced thread switch after some time in the Borkenhagen reference in no way suggests a forced configuration switch after some time.

However, Smith discloses that a function can be executed either in software or hardware (e.g. col. 7, lines 7-8). Functions executed in software are compiled into threads, and functions executed in hardware are compiled into configurations (e.g. col. 11, lines 59-63). Thus, there is a correspondence between the threads and configurations: each performs the function, albeit one in software and one in hardware. It would be readily recognized to one of ordinary skill in the art at the time of the invention that the motivation of Borkenhagen is applicable and beneficial regardless of whether functions are executed in software or hardware.

Examiner notes that Smith discloses that a typical scheduling system for a reconfigurable computer may use a time-multiplexing system in which programmable logic resources may be allocated to application functions and which may involve switching between different functions at predetermined time periods (col. 8, line 66 through col. 9, line 4). Therefore, applicant's contention that one of ordinary skill in the art would only apply multitasking teachings to threads and not to configurations is inaccurate at least because Smith already does so, though as examiner has pointed out, the correlation between the threads and configurations would be enough to motivate the combination. Nevertheless, it is Borkenhagen's multitasking policy in

particular which examiner is applying to Smith. Again, given that Smith not only correlates threads to configurations but also teaches of applying the multitasking aspect of threads to configurations, one of ordinary skill in the art at the time of the invention would recognize that Borkenhagen's specific multitasking policy would not only be applicable to threads but also to configurations, given that the goal of the policy to prevent processor hangs is applicable regardless of whether the program which is hanging is being executed in software or hardware. The execution of functions in hardware (e.g. to provide faster execution times as per Smith, col. 1, lines 46-47) does not nullify the benefit of Borkenhagen's teaching. It would be readily recognized that an explicit recitation that any general modification that one can do to threads, one can likewise do to configurations, is not necessary for the aforementioned combination to nevertheless be proper.

37. Applicant argues on page 12 that at column 8, line 66 to column 9, line 4, the Smith reference merely suggests applying a time-multiplexing system to functions, and that nowhere does the Smith reference suggest applying a time-multiplexing system to configurations.

However, Smith discloses in numerous places discloses of the correlation between functions and configurations. For example, col. 11, lines 60-63 discloses of compiling hardware functions into configuration patterns using a hardware description language compiler, and col. 9, lines 7-11 discloses of function prefetching wherein the programmable logic resource can be loaded with the configuration data before the

function is required. In other words, the invention of Smith entails functions which correlate to configuration data. As another example, col. 13, lines 31-33, discloses of a single block of configuration data making up a given function. Therefore, any extent by which Smith suggests applying a time-multiplexing system to functions would likewise apply to configuration data, as functions which are time-multiplexed and implemented via configurations consequently entails the configurations being time-multiplexed.

38. Applicant argues on page 12 that while the Smith reference may provide certain configurations of hardware that provide for operation in a manner that corresponds in its entirety to a function as a whole, the time-multiplexing is ultimately provided on a function-by-function basis, and not a configuration-by-configuration basis. Applicant provides as an example, if a configuration is usable for multiple functions, then, while the time-multiplexing may provide for interrupting a particular function, the configuration may continue to be used without reconfiguration.

However, examiner notes that the instant disclosure does not disclose of a configuration usable for multiple functions or its corresponding operation, and thus it is not necessarily the case in the hypothetical scenario that the configuration may continue to be used without reconfiguration: it could also be the case that reconfiguration occurs (e.g. because the system does not know configurations used for multiple functions are the same, or because reloading is necessary to reinitialize the configuration for use by another function, or so forth). Moreover, it is not necessarily the case that a configuration usable for multiple functions (i.e. two different functions which use the

same configuration) would even be covered by Smith. In contrast, Smith discloses of time-multiplexing functions implemented via corresponding configurations (and thus discloses of time-multiplexing configurations), which supports the modification by Borkenhagen of a specific multitasking policy to configurations in particular.

In addition, examiner generally notes that the instant disclosure does not appear to explicitly disclose of the behavior that applicant argues contrasts with Smith, as page 7, lines 6-7 only discloses that the maximum execution time of a CIW has an upper limit. There is no disclosure of "different functions" using a CIW such that a CIW is executed consecutively to carry out the different functions, for example.

39. Applicant argues on page 12 that “nowhere does the Smith reference suggest that its time-multiplexing would apply on a configuration basis, particularly where a configuration corresponds to a program loop, as required by claim 7.”

In view of the amended limitations to claim 7 which recite a program loop, examiner has incorporated the Li reference into the rejection. Specifically, Li discloses that a configuration is of an entire loop (section 1, fifth paragraph, lines 4-9, each loop has a software version and one or more hardware versions that represent different delay and area tradeoffs. The partitioning algorithm selects which loops to implement in the FPGA, and which hardware version of each loop should be used to achieve the highest application-level performance).

LI’s teaching improves system performance (Li, section 3.1, third paragraph, lines 1-5, our studies show that loops represent a significant portion of application

execution time, and yet are usually compact enough to implement in modest hardware resources. Therefore, the compiler focuses on finding the most profitable loops to extract out as hardware kernels).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Li with the invention of Smith in order to improve system performance.

40. Applicant on pages 12-13 argues that the cited references do not disclose or suggest the newly amended limitations of claim 23.

However, claim 23 has been significantly amended such that the claim is now directed to an invention that is independent or distinct from the invention originally claimed. Consequently, claim 23 and dependent claim 28 are withdrawn from consideration as being directed to a non-elected invention. For additional explanation, see the elections/restrictions section above.

41. Applicant on page 13 argues that "[t]here is no indication whatsoever that such a function compiled into a single configuration may include more than one instruction. Indeed, none of the cited references disclose or suggest that multiple instructions are executed using a single instance of a configuration."

However, the idea of a function being comprised of multiple instructions would have been obvious to one of ordinary skill in the art at the time of the invention. See for example, col. 11, lines 59-63, which discloses that software functions are compiled into

threads and hardware functions are compiled into configuration patterns. It would have been obvious to one of ordinary skill in the art at the time of the invention for a thread to be comprised of multiple instructions" (especially those compiled from "complex instructions" of col. 8, line 59). Therefore, a function which is made of multiple instructions, when executed via hardware such that a single programmable logic resource may be allocated to a single block of configuration data that makes up a given function, teaches the overall concept that multiple instructions are executed using a single instance of a configuration.

42. Applicant on pages 13-14 argues that “claims 8 and 9 ultimately depend from claim 24 and are therefore allowable for at least the same reasons set forth above in support of the patentability of claim 24.”

Consequently, examiner’s response to arguments with regard to claim 24 is likewise applicable to the arguments of claims 8 and 9.

43. Applicant on page 14 argues that “claims 12 and 13 ultimately depend from claim 24 and are therefore allowable for at least the same reasons set forth above in support of the patentability of claim 24.”

Consequently, examiner’s response to arguments with regard to claim 24 is likewise applicable to the arguments of claims 12 and 13.

44. Applicant on page 14 argues that "[c]laim 25 provides for removing a plurality of configurations in response to determining that a maximum allowed runtime of one of those configurations is exceeded. The cited references do not disclose or suggest this feature. Claim 25 and its dependent claims 26 and 27 are therefore allowable."

With regard to claim 25, the prior art of record, alone or in combination, does not disclose "responsive to determining, in the monitoring step, that the maximum allowed execution runtime of the first configuration is exceeded, removing, by the processor, the first configuration and the one or more other configurations" in the context of and in combination with the remaining limitations of the claim. Consequently, claim 25 and its dependent claims 26 and 27 are currently indicated as allowable.

45. Applicant on page 14 argues that "[c]laim 29 provides for a first configuration to trigger transmission of an entry from a stack as a call for reconfiguration. The cited references do not disclose or suggest this feature. Claim 29 and its dependent claims 30 and 31 are therefore allowable."

However, claim 29 is directed to an invention that is independent or distinct from the invention originally claimed. Consequently, claim 29 and dependent claims 30-31 are withdrawn from consideration as being directed to a non-elected invention. For additional explanation, see the elections/restrictions section above.

46. Applicant on page 14 argues that "[c]laim 32 recites features concerning the reading by a second configuration of data elements remaining in a data vector after

other data elements of the data vector were read by a prior configuration. The cited references do not disclose or suggest this feature. Claim 32 and its dependent claim 33 are therefore allowable."

With regard to claim 32, the prior art of record, alone or in combination, does not disclose "responsive to determining in the monitoring step that the maximum allowed execution runtime is exceeded, removing the first configuration and configuring the field with a second configuration prior to readout of all of the data elements of the data vector, such that a second subset of the data elements of the data vector remains unread in the memory arrangement; and subsequent to the removing of the first configuration and the configuring of the field with the second configuration, sequentially reading, by the field using the second configuration, one or more data elements of the second subset of the data elements" in the context of and in combination with the remaining limitations of the claim. Consequently, claim 32 and its dependent claim 33 are currently indicated as allowable.

Conclusion

47. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

b. Yip et al. (US 20030226056) disclose of killing a process when any of the thread timeout counts exceeds a maximum number of timeouts.

c. Paul et al. (US 7971051) disclose of FPGA configuration protection and control using hardware watchdog timer.

48. Any inquiry concerning this communication or earlier communications from the examiner should be directed to KEITH VICARY whose telephone number is (571)270-1314. The examiner can normally be reached on Monday - Friday, 7:00 a.m. - 3:30 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Keith Vicary/
Primary Examiner, Art Unit 2183